

УДК 519.68

Т.С. Щегельський, О.О. Провотар, О.І. Провотар

## АНАЛІЗ СИСТЕМ НЕЧІТКОЇ ЛОГІКИ ДЛЯ АПРОКСИМАЦІЇ НЕЧІТКИХ ФУНКЦІЙ

Розглядаються три моделі нечіткого логічного виведення: нечіткий шаблон, нечіткі нейронні мережі ANFIS і NEFPROX. Показано, що мережа ANFIS дає якісний результат апроксимації, але вимагає навчання великої кількості параметрів та складна для інтерпретації результатів. Мережа NEFPROX зручна для інтерпретації результатів. Модель на основі нечіткого шаблону дає низьку якість апроксимації та складна для інтерпретації результатів.

### Вступ

Більшість сучасних методів керування припускають наявність точної математичної моделі процесу (або системи). Як правило, таку модель важко або надто дорого отримати. З іншого боку, нечітка логіка може бути зручним інструментом для вирішення проблем побудови моделей різних процесів, якщо немає необхідності аналізувати об'єкт керування з високим ступенем деталізації. Такі моделі є наближеними, але вони можуть бути досить швидко та легко отримані. Знання в таких моделях виражають природу процесу (чи системи) у лінгвістичній інтерпретації. Це дає можливість аналізувати систему без необхідності отримання стратегії керування.

У роботі проаналізовано можливість ефективного застосування різних моделей для апроксимації нечітких функцій. Невелика кількість таких досліджень та необхідність всебічного аналізу нечітких систем зумовлює актуальність дослідження.

**Нечіткі системи.** Нечітка модель процесу зазвичай конструюється на базі знань експерта про процес (чи систему). Також часто враховуються результати експериментів, проведених над об'єктом керування. Досвід експерта, як правило, використовується для визначення структури та початкових налаштувань – змінних, функцій приналежності та нечітких правил. Після цього система налаштовується так, щоб відповідати експериментальним даним.

Існує два основних способи налаштування нечітких систем: додавання та видалення правил і корегування параметрів функцій приналежності. Перший використовується для наближеного налаштування, другий – для точного налаштування моделі.

В деяких моделях пропонується використовувати ваги правил (вага правила може інтерпретуватися як рівень його важливості) для налаштування нечіткої моделі. Нечіткий шаблон моделювання, запропонований Ягером та Файлевым в [1], використовує цей підхід.

Для визначення пар вхідних-вихідних змінних та їх нечіткого розподілу використовуються знання людини-експерта. Правильність елементарних правил, які були визначені цим розподілом, перевіряється експериментальними даними та виражається вагами правил.

Якість системи (більш точне відображення природи процесу) у значній мірі залежить від правильності вибору даних для налаштування ваг правил.

Для порівняння вибрано три нечіткі моделі: нечіткі шаблони моделювання, нечітка нейронна мережа ANFIS та мережа NEFPROX.

**Нечіткі шаблони моделювання.** Нечіткі шаблони моделювання базуються на нечіткому методі моделювання, запропонованому Ягером та Файлевым в [1], який було розширено до моделі з багатьма входами та виходами [2].

Задані пари зчитаних даних  $(x_1^k, x_2^k, \dots, x_s^k, y_1^k, y_2^k, \dots, y_t^k)$ , де  $x_1^k, x_2^k, \dots, x_s^k$  – значення входів  $U_1^k, U_2^k, \dots, U_s^k$  у момент  $k$  ( $k = (1, K)$ ),  $y_1^k, y_2^k, \dots, y_t^k$  – значення виходів  $V_1^k, V_2^k, \dots, V_s^k$  у відповідні моменти часу, відповідає елементарному правилу: якщо  $U_1$  це  $B_{i_1}^1$  AND ... AND  $U_s$  це  $B_{i_s}^s$  то  $V_1^k$  це  $D_{j_1}^1$  AND ... AND  $V_s^k$  це  $D_{j_t}^t$ , якщо всі ваги

$$\tau_{i_1}^1 = B_{i_1}^1(x_1^k), \dots, \tau_{i_s}^s = B_{i_s}^s(x_s^k),$$

$$(i_1 = (1, s_1), \dots, i_r = (1, s_s))$$

та

$$\gamma_{j_1}^1 = D_{j_1}^1(x_1^k), \dots, \gamma_{j_t}^t = D_{j_t}^t(x_t^k)$$

$$(j_1 = (1, t_1), \dots, j_r = (1, t_s))$$

не нульові. Ступінь узгодження визначається як:

$$\delta_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}(k) =$$

$$= \tau_{i_1}^1 \cdot \tau_{i_2}^2 \cdot \dots \cdot \tau_{i_s}^s \cdot \gamma_{j_1}^1 \cdot \gamma_{j_2}^2 \cdot \dots \cdot \gamma_{j_t}^t. \quad (1)$$

Одне читання вхідних-вихідних даних може мати не нульовий рівень узгодження для більш ніж одного елементарного правила. Це враховується нормалізованим рівнем узгодження  $v_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}(k)$ , який отримується нормалізацією рівня узгодженості (1) по відношенню до загального рівня узгодженості  $k$ -го читання даних з елементарними правилами

$$v_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}(k) =$$

$$= \frac{\delta_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}(k)}{\sum_{i_1=1}^{s_1} \dots \sum_{i_s=1}^{s_s} \sum_{j_1=1}^{t_1} \dots \sum_{j_t=1}^{t_t} \delta_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}(k)}. \quad (2)$$

Наступним кроком є обрахунок рівня узгодження елементарних правил по відношенню до всього вхідного набору даних

$$v_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t} = \sum_{k=1}^K v_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}(k). \quad (3)$$

Повний нормалізований рівень узгодження  $i_1 i_2 \dots i_r$ -го набору правил, який складається з правил, еквівалентних гіпотезам, визначається за формулою:

$$v_{i_1 i_2 \dots i_s} = \sum_{j_1=1}^{t_1} \sum_{j_2=1}^{t_2} \dots \sum_{j_t=1}^{t_t} v_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}. \quad (4)$$

Результатом моделювання є база правил  $R(B_{i_1}^1, \dots, B_{i_s}^s, B_{j_1}^1, \dots, D_{j_t}^t)$ , яка містить  $i_1 \cdot i_2 \cdot \dots \cdot i_s \cdot j_1 \cdot j_2 \cdot \dots \cdot j_t$  правил з вагами

$$p_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t} = \frac{v_{i_1 i_2 \dots i_s j_1 j_2 \dots j_t}}{v_{i_1 i_2 \dots i_s}}. \quad (5)$$

**Нечіткі нейронні системи.** Нейронні мережі використовуються для задач розпізнавання образів, але незручні для з'ясування питання про те, як вони таке розпізнавання здійснюють. Вони можуть автоматично набувати знання, але процес їх навчання часто відбувається досить повільно. Побудова нейронних мереж є досить простим процесом, але аналіз навченої мережі дуже складний (зазвичай навчена мережа – «чорний ящик» для користувача). При цьому будь-яку апіорну інформацію (знання експерта) для прискорення процесу її навчання в нейронну мережу ввести неможливо.

Системи з нечіткою логікою, навпаки, зручні для пояснення отримуваних за їх допомогою висновків, але не можуть автоматично набувати знання для використання їх в механізмах висновків. Необхідність розбиття універсальних множин на окремі області, як правило, обмежує кількість вхідних змінних у таких системах.

Системи з нечіткою логікою і штучні нейронні мережі еквівалентні один одному, проте, у кожному конкретному випадку в них є свої переваги та недоліки. Це лягло в основу апарата гібридних мереж, у яких висновки робляться на основі апарата нечіткої логіки, а відповідні

функції приналежності підлаштовуються з використанням алгоритмів навчання нейронних мереж, наприклад, алгоритму зворотного поширення помилки. Такі системи не тільки використовують апріорну інформацію, але можуть набувати нові знання і для користувача є логічно прозорими.

**Нечітка нейронна мережа ANFIS** (Adaptive Neuro-Fuzzy Inference System), запропонована Єнгом, визначається диференційовними операторами механізму виведення (множення та додавання), диференційовними функціями приналежності (Гауссовими) та дефузифікацією методом зваженого середнього. Дана мережа відображає механізм нечіткого виведення Тсукамото [5]. Вона може бути представлена спеціальною п'ятишаровою мережею прямого поширення. Оскільки мережа ANFIS використовує тільки диференційовні функції, для її навчання можна використовувати стандартні методи, наприклад, комбінацію алгоритму зворотного поширення похибки (метод градієнтного спуску) та методу найменших квадратів [6].

Розглянемо, наприклад, систему, що має наступну базу знань:

якщо  $x_1$  це  $L_1$  і  $x_2$  це  $L_2$  і  $x_3$  це  $L_3$ ,  
то  $y$  це  $H$ ,

якщо  $x_1$  це  $H_1$  і  $x_2$  це  $H_2$  і  $x_3$  це  $L_3$ ,  
то  $y$  це  $M$ ,

якщо  $x_1$  це  $H_1$  і  $x_2$  це  $H_2$  і  $x_3$  це  $H_3$ ,  
то  $y$  це  $S$ ,

де  $x_1, x_2, x_3$  – вхідні змінні,  $y$  – вихід системи,  $L_1, L_2, L_3, H_1, H_2, H_3, H, M, S$  – деякі нечіткі множини з функціями приналежності сигмоїдного типу:

$$\begin{aligned} L_j(t) &= \frac{1}{1 + e^{b_j(t-c_j)}}, \\ H_j(t) &= \frac{1}{1 + e^{-b_j(t-c_j)}}, j = 1, 2, 3, \\ H(t) &= \frac{1}{1 + e^{-b_4(t-c_4+c_5)}}, \\ M(t) &= \frac{1}{1 + e^{-b_4(t-c_4)}}, \\ S(t) &= \frac{1}{1 + e^{b_4(t-c_4)}}. \end{aligned} \quad (6)$$

Для визначення вихідної змінної використовується алгоритм виведення Тсукамото:

1) підраховуються значення істинності передумов для кожного правила:

$$\begin{aligned} \alpha_1 &= L_1(a_1) \wedge L_2(a_2) \wedge L_3(a_3), \\ \alpha_2 &= H_1(a_1) \wedge H_2(a_2) \wedge L_3(a_3), \\ \alpha_3 &= H_1(a_1) \wedge H_2(a_2) \wedge H_3(a_3), \end{aligned} \quad (7)$$

де в цьому випадку  $a_1, a_2, a_3$  – поточні значення входів системи;

2) для кожного правила визначаються часткові виходи:

$$\begin{aligned} z_1 &= B^{-1}(\alpha_1) = c_4 + c_5 + \frac{1}{b_4} \ln \frac{1 - \alpha_1}{\alpha_1}, \\ z_2 &= B^{-1}(\alpha_2) = c_4 + \frac{1}{b_4} \ln \frac{1 - \alpha_2}{\alpha_2}, \\ z_3 &= B^{-1}(\alpha_3) = c_4 + \frac{1}{b_4} \ln \frac{1 - \alpha_3}{\alpha_3}; \end{aligned} \quad (8)$$

3) знаходиться загальний вихід системи:

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3}{\alpha_1 + \alpha_2 + \alpha_3}. \quad (9)$$

Гібридна нейронна мережа, яка відображає наведений механізм виведення, показана на рис. 1.

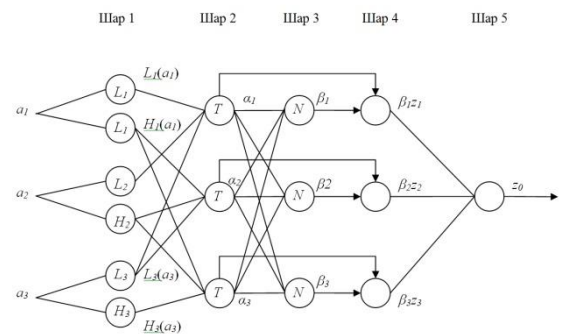


Рис. 1. Структура мережі ANFIS

Ця мережа може бути описана таким чином:

Шар 1. Виходи вузлів цього шару представляють собою значення функцій приналежності при конкретних (заданих) значеннях входів.

Шар 2. Виходами нейронів цього шару є ступені істинності передумов кожного правила бази знань системи, які обчислюють за формулами (7). Всі нейрони цього шару позначені буквою  $T$ . Це означає, що вони можуть реалізовувати довільну  $t$ -норму.

Шар 3. Нейрони цього шару (позначені буквою  $N$ ) обчислюють величини:

$$\begin{aligned}\beta_1 &= \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \\ \beta_2 &= \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}, \\ \beta_3 &= \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}.\end{aligned}\quad (10)$$

Шар 4. Нейрони цього шару виконують операції:

$$\begin{aligned}\beta_1 z_1 &= \beta_1 H^{-1}(a_1), \\ \beta_2 z_2 &= \beta_2 M^{-1}(a_2), \\ \beta_3 z_3 &= \beta_3 S^{-1}(a_3).\end{aligned}\quad (11)$$

Шар 5. Єдиний нейрон цього шару обраховує вихід мережі:

$$z_0 = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3. \quad (12)$$

Коригування параметрів системи відбувається за правилом градієнтного спуску. Так, наприклад, налаштування коефіцієнтів  $b_4$ ,  $c_4$  і  $c_5$  відбувається за формулами:

$$\begin{aligned}b_4 &= b_4 - \eta \frac{\partial E_k}{\partial b_4} = \\ b_4 - \frac{\eta}{b_4^2} \delta_k \frac{a_1 + a_2 - a_3}{a_1 + a_2 + a_3}, \\ c_4 &= c_4 - \eta \frac{\partial E_k}{\partial c_4} = \\ &= c_4 + \eta \delta_k \frac{a_1 + a_2 + a_3}{a_1 + a_2 + a_3} = c_4 + \eta \delta_k, \\ c_5 &= c_5 - \eta \frac{\partial E_k}{\partial b_4} = c_5 + \eta \delta_k \frac{a_1}{a_1 + a_2 + a_3},\end{aligned}\quad (13)$$

де  $\delta_k = y^k - o^k$ .

Відповідні вирази можуть бути отримані й для решти коефіцієнтів.

**Мережа NEFPROX** базується на нечіткому перцептроні [3]. Елементи мережі використовують  $t$ -норми або  $s$ -норми замість активаційних функцій, які зазвичай використовуються в нейронних мережах. Перший шар представляє вхідні змінні, другий (прихований) – нечіткі правила, третій – вихідні змінні. Нечіткі множини кодуються як колекції ваг  $W(x, R)/W(R, y)$ . Ваги, які відповідають тим самим функціям приналежності, мають однакові значення.

Нечіткі системи мають мало обмежень – практично можуть використовуватись усі  $s$ - та  $t$ -норми та всі типи функцій приналежності, також середнє значення в якості дифузіфікатора. Система може бути ініціалізована апіорними знаннями експерта. Метод градієнтного спуску не може бути використаний, оскільки  $t$ - та  $s$ -норми не є диференційованими функціями. Замість нього використовують просту евристичну процедуру, яка складається із двох кроків: навчання нечітких правил (навчання структури мережі) та навчання нечітких множин (навчання параметрів), (рис. 2).

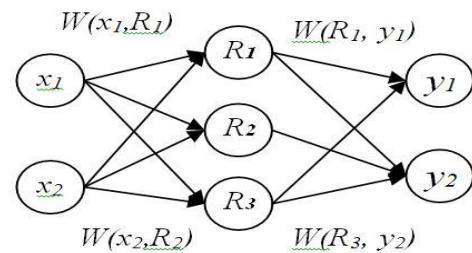


Рис. 2. Структура мережі NEFPROX

Ітераційний алгоритм навчання правил мережі створює базу правил (починаючи з порожньої множини правил, якщо потрібно) послідовним додаванням нових правил. Для кожної навчальної пари нечітка множина, яка забезпечує найвищий ступінь приналежності, використовується для конструювання нового правила. Під час другої стадії база правил оптимізується шляхом

зміни наслідків для відповідних функцій приналежності, якщо це потрібно.

Результатом навчання параметрів мережі є корегування функцій приналежності. На основі роботи правил, алгоритм визначає необхідність зменшення чи збільшення ступеня активації правила і визначає нечітку множину, яка відповідає за активацію даного правила.

**Система.** Для порівняння та оцінки моделей нечіткої логіки було використано просту нечітку систему з одним входом та одним виходом. Вхідна змінна  $x = [0, 10]$  та вихідна змінна  $y = [-5, 5]$  розділені відповідно на 6 та 5 функцій приналежності  $mf_i$ . На рис. 3 показано ці функції приналежності, нечіткі правила та відношення входом та виходом результуючої системи.

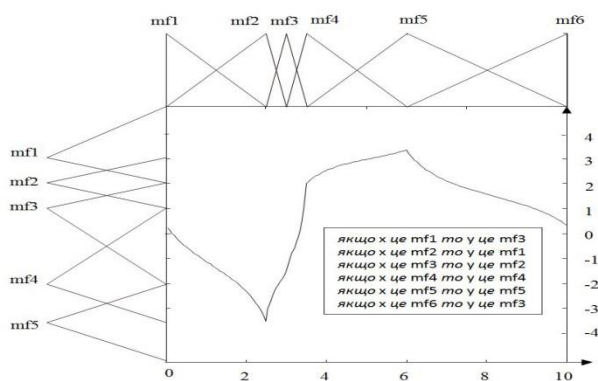


Рис. 3. Нечітка система

Функції приналежності та базу правил вибрано таким чином, щоб побудувати функцію, яку б було важко апроксимувати.

#### Порівняння та оцінка моделей.

Першою була змодельована система методом Янгера-Файлева із використанням:

- 1) симетричних трикутних функцій приналежності зі ступенем перекривання 0.5;
- 2) нечіткого поділу оригінальної системи.

Результуючі моделі (правила, ваги яких були менші за 0.1, не вносились в таблиці для більшої наочності результату) приведені в табл. 1 та 2 відповідно.

Таблиця 1

Вхід/ Вихід	f1	f2	f3	f4	f5
mf1		.414	.568		
mf2	.248	.562	.149		
mf3		.111	.241	.309	.437
mf4				.331	.669
mf5			.124	.789	
mf6			.487	.513	

Таблиця 2

Вхід/ Вихід	f1	f2	f3	f4	f5
mf1		.243	.377		
mf2	.603	.658			
mf3		.717	.241		
mf4			.103	.330	.520
mf5			.140	.342	.516
mf6			.644	.290	

Результат роботи моделі на вхідному проміжку  $[0, 10]$  у порівнянні з оригінальною системою показаний на рис. 4. Система на основі цієї моделі в обох випадках демонструє низьку якість апроксимації. Ще одним недоліком моделі є необхідність застосування ваг для кожного нечіткого правила. Це порушує семантику нечіткої системи та робить її інтерпретацію досить складною. При аналізі моделі бралися до уваги тільки правила з великими вагами. Правила з малими вагами можуть призводити до хибних висновків.



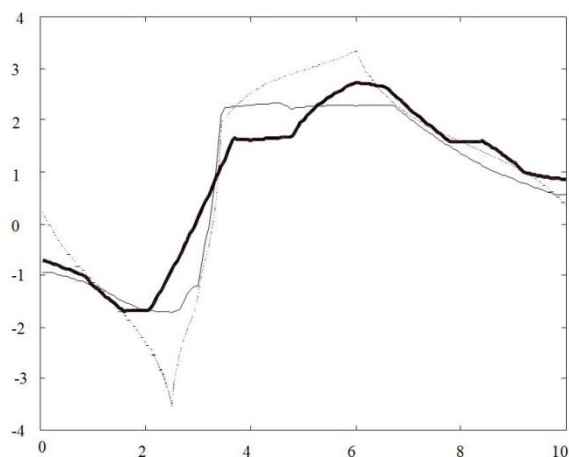


Рис. 4. Моделювання методом Янгера-Файлева

Для мережі ANIFS визначено шість апіорних функцій приналежності. Для цього використано метод субтрактивної кластеризації (рис. 5).

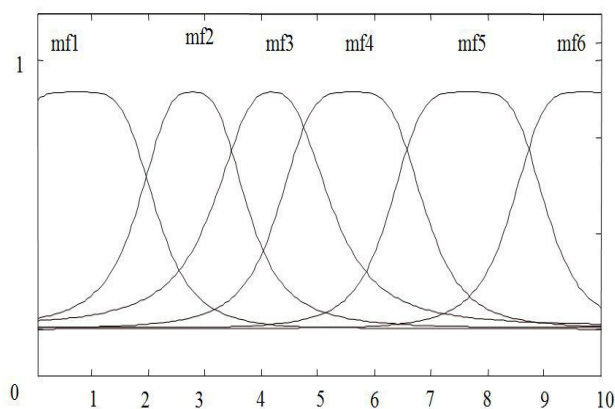


Рис. 5. Апіорні функції приналежності мережі ANIFS

Результат навчання мережі показаний на рис. 6.

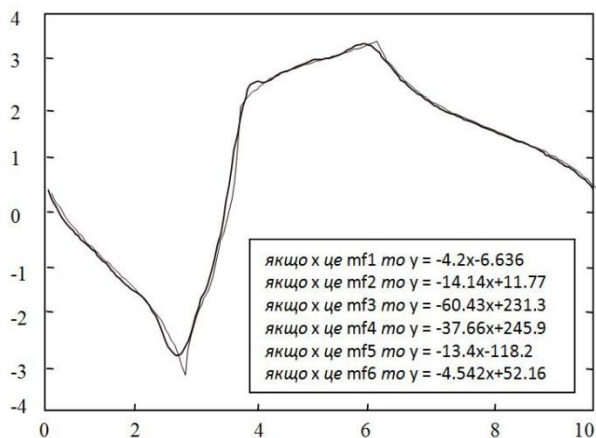


Рис. 6. Моделювання за допомогою мережі ANIFS

Мережа ANIFS у поєднанні з субтрактивною кластеризацією дає високу якість апроксимації. Це можна пояснити великою кількістю параметрів, що підлягають налаштуванню та аналітичним правилами навчання. Інтерпретація правил Сугено є досить складною, тому використовувати цю мережу доцільно тоді, коли якість роботи більш важлива, ніж необхідність розуміння природи процесу.

Результат роботи мережі NEFPROX показано на рис. 7. Аналогічно до нечіткого шаблону, цей метод не дозволяє відтворити оригінальну модель, але в даному випадку немає потреби використовувати додаткові ваги та відклик набагато ближчий до оригінальної системи.

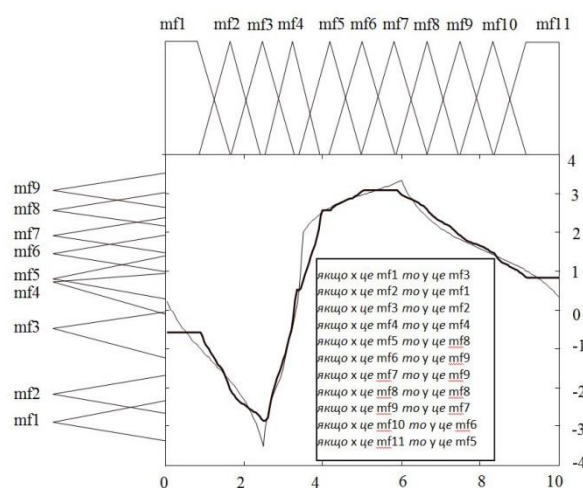


Рис. 7. Моделювання за допомогою мережі NEFPROX

Аналогічно до ANIFS, якість роботи мережі NEFPROX сильно залежить від кількості функцій приналежності, які мають бути визначені до початку навчання. Збільшення кількості функцій приналежності зумовлює кращий результат.

## Висновки

Розглянуті моделі нечіткого логічного виведення по-різному виконують завдання апроксимації нечітких функцій. Модель ANIFS дає якісний результат апроксимації, але потребує навчання великої кількості параметрів та складна при інтерпретації. Модель NEFPROX не повністю відтворює оригінальну модель, але зручна для інтерпретації результатів.

1. *Yager R. and Filev D.* Essentials of Fuzzy Modeling and Control // John Wiley and Sons, New York, 1994.
2. *Riid A. and Rüstern E.* Fuzzy modeling and control of fed-batch fermentation // Proc. 9th Int. Symp. on System-Modelling-Control, Zakopane, Poland, 1998.
3. *Nauck D., Klawonn F. and Kruse R.* Foundations of Neuro-fuzzy Systems // John Wiley and Sons, New York, 1997.
4. *Круглов В.В.* Искусственные нейронные сети // Теория и практика. – М. : Горячая линия – Телеком, 2002. – 382 с.
5. *Осовский С.* Нейронные сети для обработки информации: пер. с польск. И.Д. Рудинского. – М. : Финансы и статистика, 2002. – 344 с.
6. *Рудковская Д., Пилитинский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы : пер. с польск. И.Д. Рудинского. – М.: Горяч. линия – Телеком, 2006. – 452 с.

Одержано 21.10.2011

***Про авторів:***

*Щегельський Тарас Сергійович,*  
аспірант,

*Проватар Ольга Олександрівна,*  
аспірантка,

*Проватар Олександр Іванович,*  
доктор фізико-математичних наук,  
професор.

***Місце роботи авторів:***

Волинський національний університет  
імені Лесі Українки  
Тел. 066 627 3357,  
e-mail: [shchegelskij@gmail.com](mailto:shchegelskij@gmail.com)

Інститут кібернетики  
імені В.М. Глушкова НАН України  
Тел. 063 034 2230,  
e-mail: [aprowata@unicyb.kiev.ua](mailto:aprowata@unicyb.kiev.ua)

Київський національний університет  
імені Тараса Шевченка  
Університет міста Жешув (Польща)  
Тел. 050 444 1735,  
e-mail: [aprowata@unicyb.kiev.ua](mailto:aprowata@unicyb.kiev.ua)